Natural Language Processing for Social Media Analytics

Caroline Brun
Naver Labs Europe
caroline.brun@naverlabs.com
Outline

• Social Media Analytics: Introduction
  • Overview
  • Applications
  • Challenges

• NLP and Social Media Texts
  • Normalization
  • POS Tagging

• Sentiment Analysis
  • Reminder on text classification and evaluation measures
  • Affective lexicons
  • Sentiment analysis: Introduction
  • Document Level sentiment analysis
  • Aspect Based Sentiment analysis

• Fake news and stance detection
Social Media Analytics: Introduction
vintage social networking

LinkedIn
Pinterest
YouTube
foursquare
Instagram
Imgur
WordPress
Twitter
Facebook
reddit
Skype
Tumblr

http://wronghands1.wordpress.com
© John Atkinson, Wrong Hands
Social Media: Big Change

• Web is no longer a static library that people passively browse

• Web is now a place where people:
  • Consume and create content
  • Interact with other people:
    • Social networking sites: Facebook, Twitter, Wikipedia...
    • Blogs, wikis, news, online forums, online reviews
Number of Global Social Network Users 2010-2021*
In Billion

Social Media Global Revenue 2013-2019*
In Billion Euros

*source: statista
Social Media: Rich and Big Data

• Rich and big data:
  • Billions users, billions contents
  • **Textual**, Multimedia (image, videos, etc.)
    • But text is a core data type
  • Billions of connections
  • Behaviors, preferences, trends...

• Data is open and easy to access
  • It’s “easy” to get data from Social Media
  • Datasets
  • Developers APIs
  • Spidering the Web
Social Media: Opportunities

• Any user can share and contribute content, express opinions, links to others
• One can data-mine opinions and behaviors of millions of users
• In contrast to survey/self reports
• A probe to:
  • Real human behavior
  • Real human opinions
  • Real human language use
• Gain insights into:
  • Personality traits
  • Marketing analytics
  • Product sentiments
  • People’s opinions about a wide range of topics

➤ Great opportunity for NLP!
Social Media Analytics

- **Capture**
  - Gather data from various sources
  - Preprocess the data
  - Extract pertinent information from the data

- **Understand**
  - Remove noisy data (optional)
  - Perform advanced analytics: opinion mining and sentiment analysis, topic modeling, social network analysis, and trend analysis

- **Present**
  - Summarize and evaluate the findings from the understand stage
  - Present the findings

- **Network analysis + machine learning + natural language processing (NLP) + statistics**
Applications (1)

• Health care applications
  • Prediction of depression, suicidal tendencies

• Business and management
  • Brand awareness, service improvement, advertising/marketing strategies, identifying influencers

• Financial applications
  • Determine market sentiment, news data for trading
  • Twitter data to forecast NASDAQ stock prices
Applications (2)

• Predicting voting intentions
  • Monitoring public perception on political candidates, election campaigns and announcements

• Security and defense applications
  • Monitoring terrorist activities, crimes, threats, etc.
    • Detecting radicalism
    • Detecting potential criminals

• News
  • Monitoring, analysis, and aggregation of events from user-generated content

• Disaster forecasting, crisis monitoring

• NLP-based user modelling (“profiling”)
Social Media: Many R&D Topics for NLP

- Geo-location detection
- Event and topic detection
- Entity linking and disambiguation
- Summarization
- Personality profile detection
- Influencer detection
- Sentiment analysis: opinion mining, emotion analysis
- Fake news detection & Stance detection
Example 1: User Profiling

Delighted I kept my Xmas vouchers - Happy Friday to me 😊 #shopping

Yesterday's look-my new obsession is this Givenchy fur coat! Wolford sheer turtleneck, Proenza skirt & Givenchy boots

We've already tripled wind energy in America, but there's more we can do.

Two giant planets may cruise unseen beyond Pluto - space - June 2014 - New Scientist: newsscientist.com/article/dn2571

Source: Volkova et. al Tutorial on Social Media Predictive Analytics @NAACL 2015
Example 2: Health

- **Negative emotions** (anger, stress, fatigue…) in tweets are associated with higher heart disease risk.
- **Positive emotions** like excitement and optimism are associated with lower risk.

Source: World Well-Being Project @ University of Pennsylvania
Many Challenges

• Restrictions imposed by websites on data collections (tweeter, customer review sites, Facebook), General Data Protection Regulation (GDPR)

• Spread of fake news, unsubstantiated rumors, fake reviews

• Nonhuman: social bots

• Analysis may misrepresent the real world

• Distortion of human behavior: social platforms serve specific, practical purpose -- not necessarily to represent social behavior
  • e.g. Facebook versus Linkedin

• Social Media are dynamic in nature and their sheer size pose serious computing problems

• Social Media yield **unstructured noisy data**
NLP and Social Media Texts
Properties of Social Media **Textual** Data

- Key properties:
  - Social, real time, geo-spatially coded, emotions, neologisms, credibility/rumors
- Non structured text in many formats
  - What’s a sentence?
- Written by different people in many languages and styles
- Written in everyday languages, generally by non-professional writers
- Sources in thousand of place on the WWW
Social Media Texts: a Foe of NLP?

• ++: Big Data “easily” accessible, opportunity for NLP
• But: Are NLP methods suited for social media analysis?
• **Free form nature of language on social media = “Noisy Input”**
  • Spelling inconsistencies
  • Dialectal peculiarities of a given user
  • Adoption of new terms (lexical creativity)
  • Regular violation of grammar rules/norms
  • Often short text (e.g. microblogs like twitter): lack of context
  • Mix of languages
July 4 we celebrate freedom and liberty that started w/ the greatest document ever written, Declaration of Independence. SO have happy Independence Day. #IPARADE
Awesome spot! Even at 5:45pm place was packed. Everyone looks happy (that's always a good sign) Bestie took me here for bday dinn just now. Parking is a bit on/off but that's given in sf on Thursday night. We sat at the lively bar and enjoyed it plenty. Luv the open industrial concept!! Gf introduced me to their famous donut, great fluffy texture. Uni pasta. Foie gras torchon. Flan with caviar topping. Cavier buttermilk biscuit. Hamachi collar. Aged beef wontons with chili sauce. Oyster with wasabi. Liver mousse with kimchi. Just the names of what we ate makes me hungry. Visual affects!! Good looking servicemen...Thanku for the birthday celebration experience!
Lexical Analysis

Analysis of the rate of out of vocabulary (OOV) words on a set of social media and standard corpus

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Word length</th>
<th>Sentence length</th>
<th>%OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter-1</td>
<td>3.8±2.4</td>
<td>9.2±6.4</td>
<td>24.6</td>
</tr>
<tr>
<td>Twitter-2</td>
<td>3.8±2.4</td>
<td>9.0±6.3</td>
<td>24.0</td>
</tr>
<tr>
<td>Comments</td>
<td>3.9±3.2</td>
<td>10.5±10.1</td>
<td>19.8</td>
</tr>
<tr>
<td>Forums</td>
<td>3.8±2.3</td>
<td>14.2±12.7</td>
<td>18.1</td>
</tr>
<tr>
<td>Blogs</td>
<td>4.1±2.8</td>
<td>18.5±24.8</td>
<td>20.6</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>4.5±2.8</td>
<td>21.9±16.2</td>
<td>19.0</td>
</tr>
<tr>
<td>BNC</td>
<td>4.3±2.8</td>
<td>19.8±14.5</td>
<td>16.9</td>
</tr>
</tbody>
</table>

source: [Baldwin et al 2013]: “How Noisy Social Media text, how different social media sources”, IJCNLP 2013
Standard NLP tools Perform Poorly on Social Media Textual

- **Tokenization**
  Segmenting a sequence of character codes into a sequence of basic tokens (wordforms, punctuation symbols, numbers)

- **Lemmatization/stemming**
  Assigning to each token its base form (lemma) and/or root form

- **Part-of-speech (POS) tagging**
  Assigning to each token its part-of-speech tag (etiqueta de la categoría gramatical)

- **Parsing (Syntactic analysis)**
  Building syntactic structures of sentences (words are grouped into syntactic constituents and/or interconnected with grammatical relations)

- **Semantic processing**
  Semantic Role Labeling (SRL)
  Abstract Meaning Representation (AMR)
  Coreference resolution
Example: POS Tagging and Parsing of Well-Formed Text

You must be talking about the paper, but I was thinking movies.
Example: POS Tagging and Parsing of Twitter Text

```
"u must be talkin bout the paper, but I was thinkin movies."
```

```
S
  /\   \\
S -\ CC -\ S
  \  |  /  \\
  NP VP NP VP
     /  \
    /   \
   VP VP
   /   /
  be VBN S
     /  \
    VBP VP
   /    /
  talkin VBG NP
   /     \
  "bout DT NN
  "the paper
```
Possible ways to tackle the problem(s)

• “Adapt” data to NLP tools through preprocessing of various forms
  • Lexical normalization (example 1)
• “Adapt” the NLP tools to data through “domain” adaptation
  • POS tagging (example 2)
  • Chunking
  • Named Entity Recognition
  • Etc.
Adapt Data to NLP: Lexical Normalization

→ Translate expressions into their canonical form, consistent with dictionary and grammar
Adapt Data to NLP: Lexical Normalization

“Lexical Normalization of Short Text Messages: Makn Sens a #twitter”, Han & Baldwin, ACL 2011

• Definition of “lexical normalization”:
  • relative to some standard tokenization and standard lexicon
  • consider only OOV tokens as candidates for normalization
  • allow only one-to-one (IV) word substitutions

i left ACL cus im sickk ! Yuu better be /their/ tmrw
⇓⇓⇓⇓
i left ACL because im sick ! You better be /their/ tomorrow

• Assumptions:
  • No normalization of IV tokens, e.g., their (“there”)
  • No normalization to multiword tokens, e.g., ignore ttyl (“talk to you later”)
  • Ignore Twitter-specific entities, e.g., @obama, #mandela, bit.ly/1iRQeMe
  • Assumes a unique correct “norm” for each token (reference lexicon)
Adapt Data to NLP: Lexical Normalization

“Lexical Normalization of Short Text Messages: Makn Sens a #twitter”, Han & Baldwin, ACL 2011

• Normalizing text brings out of vocabulary rates down to more conventional levels

• Major challenge: lack of annotated data
  • Unsupervised or semi supervised methods
  • Evaluated on small hand annotated datasets

• Token-based approaches
  • Confusion set generation : find correction candidates
  • Non standard word detection
  • Normalization = selection of the best candidate
An Example of Token-Based Approaches

... crush da redberry b4 da water ...

before four be bore ...

\[ \begin{align*}
\text{crush da redberry} & \quad (\text{four be bore}) \\
(\text{-3 -2 -1}) & \quad (+1 +2)
\end{align*} \]

\[ \text{before} \quad \text{da water} \]

\[ \text{Ill-formed word detector} \quad \text{Yes or No} \]

candidates and context

<table>
<thead>
<tr>
<th>before</th>
<th>crush</th>
<th>-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>four</td>
<td>da</td>
<td>-2</td>
</tr>
<tr>
<td>be</td>
<td>redberry</td>
<td>-1</td>
</tr>
<tr>
<td>bore</td>
<td>da</td>
<td>+1</td>
</tr>
</tbody>
</table>

before 1.2

<table>
<thead>
<tr>
<th>four</th>
<th>0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>be</td>
<td>0.1</td>
</tr>
<tr>
<td>bore</td>
<td>0.2</td>
</tr>
</tbody>
</table>
An Example of Token-Based Approaches

• Candidate Set Generation:
  • Generation via edit distance over letters and phonetic normalization: match words which sound similar, using Soundex or metapone algorithm [Philips, 2000]), e.g. “love” and “laugh” are candidates for “luv”

• Ill formed word detection based on “context fitness”
  • correct words should fit better with context than substitution candidates
  • incorrect words should fit worse that substitution candidates

<table>
<thead>
<tr>
<th>Ill-formed word in text snippet</th>
<th>Candidate</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>but I was thinkin movies.</td>
<td>(thinking, ...)</td>
<td>dobj(thinking, movies)</td>
</tr>
<tr>
<td>article poster by ruderrobb : there was</td>
<td>(rattrap, ...)</td>
<td>–</td>
</tr>
</tbody>
</table>

⇒ A SVM classifier is trained based on syntactic dependencies, to indicate candidate context fitness
Feature Representation Using Dependencies

- Build a dependency bank from existing corpora
- Dependency tuple represented word pair + positional index (without the dependency type)

**Stanford Parser applied on NYT corpora**

*One obvious difference is the way they look, ....*

- num(difference:3, One:1)
- amod(difference:3, obvious:2)
- nsubj(way:6, difference:3)
- cop(way:6, is:4)
- det(way:6, the:5)
- dobj(look:8, way:6)
- nsubj(look:8, they:7)
- rcmad(way:6, look:8)

**Dependency Bank**

(way,difference,3)
(look,way,2)
SVM Training Data Generation

• Use dependency bank directly as positive features
• Automatically generate negative dependency features by replacing the target word with highly-ranked confusion candidates
Detecting Ill-Formed Words

- OOV words with replacement candidates fitting the context (i.e. positive classification outputs) are probably ill-formed words

- lookin is selected as an ill-formed word

- Best candidate selection: If several candidates, selection of the best one with word similarity measure and dependency classification score
## Evaluation Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Evaluation</th>
<th>NC</th>
<th>MT</th>
<th>DL</th>
<th>WS</th>
<th>CS</th>
<th>WS+CS</th>
<th>DL+WS+CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>Precision</td>
<td>0.465</td>
<td>—</td>
<td>0.927</td>
<td>0.521</td>
<td>0.116</td>
<td>0.532</td>
<td>0.756</td>
</tr>
<tr>
<td>SMS</td>
<td>Recall</td>
<td>0.464</td>
<td>—</td>
<td>0.597</td>
<td>0.520</td>
<td>0.116</td>
<td>0.531</td>
<td>0.754</td>
</tr>
<tr>
<td>SMS</td>
<td>F-score</td>
<td>0.464</td>
<td>—</td>
<td>0.726</td>
<td>0.520</td>
<td>0.116</td>
<td>0.531</td>
<td>0.755</td>
</tr>
<tr>
<td>Twitter</td>
<td>Precision</td>
<td>0.452</td>
<td>—</td>
<td>0.961</td>
<td>0.551</td>
<td>0.194</td>
<td>0.571</td>
<td>0.753</td>
</tr>
<tr>
<td>Twitter</td>
<td>Recall</td>
<td>0.452</td>
<td>—</td>
<td>0.460</td>
<td>0.551</td>
<td>0.194</td>
<td>0.571</td>
<td>0.753</td>
</tr>
<tr>
<td>Twitter</td>
<td>F-score</td>
<td>0.452</td>
<td>—</td>
<td>0.622</td>
<td>0.551</td>
<td>0.194</td>
<td>0.571</td>
<td>0.753</td>
</tr>
</tbody>
</table>

Table 3: Candidate selection effectiveness on different datasets (*NC* = noisy channel model (Cook and Stevenson, 2009); *MT* = SMT (Aw et al., 2006); *DL* = dictionary lookup; *WS* = word similarity; *CS* = context support)
Adapt NLP to data: CMU Twitter POS tagger

• Ref.: “Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiment” [Gimpel et al. 2011], ACL 2011, Portland, USA

• Plan:
  • Large team of annotators
  • Simple, carefully-designed annotation scheme
  • Features leveraging existing resources (treebanks) and unannotated data

• Outcome
  • Tag set for twitter
  • 1827 annotated English tweets
  • POS Tagger with ~90% accuracy
Data on twitter

- Multi word abbreviations
- Non standard spelling
- Hashtags

Also: at-mentions, URLs, emoticons, symbols, typos
Method

• Start with coarse set of Penn Treebank tags
  
  Common noun  pronoun  proper noun
  Determiner  preposition  verb particle
  Verb  adjective  numeral
  Adverb  interjection  punctuation
  Coordinating Conjunction  predeterminer (there)

• Add Twitter-specific tags
Penn Treebank tokenization is unsuitable for Twitter

@user1 OMG ur from PA ? i am too (: where abouts ?

you’re

I’m going to

@user2 ima get me a flip phone for real
Penn Treebank tokenization is unsuitable for Twitter

@user1 OMG ur from PA ? i am too (: where abouts ?

you’re

I’m going to

@user2 ima get me a flip phone for real

Nominal + verbal
Penn Treebank tokenization is unsuitable for Twitter

Nominal + verbal

@user1 OMG ur from PA ? i am too (: where abouts ?

you’re

I’m going to

@user2 ima get me a flip phone for real

Nominal + verbal

➡️ Solution: don’t try to tokenize these, instead introduce compound tags
Twitter-specific tags

• Hashtag
• At-mention
• URL/email address
• Emoticon
• Twitter discourse marker
• Other (multitword abbreviations, symbols, garbage)
Hashtags

Twitter hashtags are sometimes used as ordinary words (35% of the time) and other times as topic markers.

Innovative, but traditional, too! Another fun one to watch on the iPad!
http://bit.ly/ @user1 #utcd2 #utpol #tcot

Hashtag

⇒ hashtag only used for topic markers

Proper noun
Twitter Discourse Marker

Retweet construction:

RT @user1: I never bought candy bars from those kids on my doorstep so I guess they’re all in gangs now.

RT @user2: LMBO! This man filed an EMERGENCY Motion for Continuance on account of the Rangers game tonight. 😂Wow lmao
Data Annotation

• Resulting target: 25 POS tags
• 17 annotators (2-20 hours annotating from Stanford Tagger output)
• +2 annotators for correction and standardization

• Quality measurement
  • Inter annotator agreement 92.2%
  • Cohen’s kappa: 0.914: measure of agreement taking into account chance of agreement

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}, \quad \frac{p_o}{p_e},$$

- $p_o$ is the relative observed agreement among raters (~accuracy)
- $p_e$ is the hypothetical probability of chance agreement

K=1 if complete agreement
Tagging experiments

• 1,827 annotated tweets
  • 1000 for training
  • 327 for development
  • 500 for testing (OOV rate=30%)

• Systems
  • Stanford tagger (retrained on these data)
  • Their own baseline CRF tagger
  • Their own baseline tagger augmented with Twitter-specific features
Twitter specific features

• Orthographic features: regular expressions to capture at-mentions, hashtags and URLs (#.*, http//.*, ...)

• Distributional similarity features
  • Distributional features from the successor and predecessor probabilities for the 10 000 most common terms
  • Computed using 134K unannotated tweets

• Phonetic normalization features: Metaphone algorithm
  • Phonetic normalization rules: match words which sound similar (~ soundex)

• Tag Dictionary features
  • One feature for each tag a word occurs with in the Penn Treebank, with its frequency rank
Tagging experiments: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford tagger</td>
<td>85.85</td>
</tr>
<tr>
<td>Their CRF baseline tagger</td>
<td>83.38</td>
</tr>
<tr>
<td>Their CRF baseline tagger with all features</td>
<td>89.37</td>
</tr>
<tr>
<td>Inter-annotator agreement</td>
<td>92.2</td>
</tr>
</tbody>
</table>
Summary: NLP tools and Noisy Input

• Lexical Normalization
  • Token based approaches
  • Distributional similarity approaches

• NLP Tools adaptation (POS, NER, Parsing)
  • CMU ARK TweetNLP [http://www.ark.cs.cmu.edu/TweetNLP/](http://www.ark.cs.cmu.edu/TweetNLP/) : Twokenizer, POS tagger, TweeboParser
  • Gate Twitter Pos Tagger [https://gate.ac.uk/wiki/twitter-postagger.html](https://gate.ac.uk/wiki/twitter-postagger.html)

• Active research domain : Workshop on Noisy User-generated Text (W-NUT 2015, 2016, 2017, 2018, ...) & Shared tasks on Twitter Lexical Normalization & Named Entity Recognition on Social Medias
Sentiment Analysis
Outline

• Reminder on text classification and evaluation measures
• Affective lexicons
• Sentiment Analysis: Introduction
• Document Level sentiment analysis
• Aspect Based Sentiment analysis
Reminder: Text Classification and Evaluation Measures
Supervised vs. Unsupervised Learning

• **Unsupervised Learning**
  - The class label of training data is unknown
  - Given a set of measurement, observation, etc., with the aim of establishing the existence of classes or clusters in the data

• **Supervised Learning**
  - Supervision: the training data (observations, measurement) are accompanied by **labels** indicating the class of the observations
  - New data is classified based on the training set
Text Classification Definition

- **Supervised Learning**: existence of labeled training documents
- The classifier (model usage=prediction)
  - Input: a document $x$
  - Output: a predicted class $y$ from some fixed set of labels $y_1, y_2, \ldots, y_n$
- The learner (model construction)
  - Input: set of $m$ labeled documents $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$
  - Output: a learned classifier $f: x \rightarrow y$
**Learner:**
- From examples of a target concept, build a model to explain the concepts.

**Classifier:**
- Classification model used for classifying future or unknown case.
- Estimate accuracy of the model.
- Performance depends on how accurately the hypothesis explains the examples.
Some Classification Algorithms

- Logistic regression
- Naive Bayes
- Maximum Entropy
- Support Vector Machines
- Decision Trees
- K Nearest Neighbor
- Neural Networks
  - Perceptron, multilayer perceptron
  - Recurrent Neural Network (LSTMs)
  - Convolutional Neural Networks

Useful links:
Vector Space Model

- Tokenization
- Word Normalization:
  - Stemming
  - Lemmatization
- Word scoring:
  - Number of occurrences
  - Presence or absence
  - Tf-idf

The Bag of Words Representation
The Bag-of-Words Model (BOW)

**bag-of-words** model: simplifying representation

A text (sentence, document) is represented as the bag (multiset) of its *words*, disregarding grammar and even *word* order but keeping multiplicity

→ Baseline representation of documents (i.e. feature generation) for classification

• Simple extension: **N-gram model**
  • N-gram: contiguous sequences of N words
  • Bigram: *the cat, the dog, dog bark, ..*
  • Trigram: *the dog bark, bark in the, in the garden*
  → bag-of-word model = special case of the n-gram model, with n=1 (unigram)
Reminder: Tokenization

Chopping up sentences into smaller pieces (words or tokens)

• **Choices for delimiters:**
  • Usually white space (for European Languages)
    “We’re going to Barcelona”, “We’re going to the Hague”
  • Apostrophes
    “George’s phone” ⇒ “George” and “phone”
    I’m, we’re, they’re ⇒ I am, we are and they are.
    It can also be used as a quotation mark
  • Is far more complex for other languages e.g. Asian languages like Chinese:
    电脑是新的

• **Punctuation:**
  • Sometimes removed for topic classification
  • Useful for sentiment classification ("!!!", "!", ")
Reminder: Word Stemming/Lemmatization

• Reduction of each word to its base (or stem) form (by chopping of the affixes or by morphological lemmatization)
  • Stemming: cats ➔ cat, playing ➔ play
  • Lemmatization: were, was, is ➔ be

• Capital letters should be normalized to lowercase; unless for proper nouns

• Multiple writing: USA, U.S.A.

• Ambiguity: US, us

• Harder for Social Media Input (c.f. beginning of the course)

• Stop words (the, a, ...) removal can be applied
Bag-of-Words Model (BOW): scoring functions

- **Presence of absence** of the word: binary score (0 or 1)
- **Word frequency**: nb of times the word appears in the document
  - Drawback: gives importance to high frequency words like “the”, “a”, “to” and less importance to content words
- **Term frequency–inverse document frequency** (tf-idf)
  - **tf**: term frequency: \( tf(t,d) = \frac{\text{number of times } t \text{ appear in } d}{\text{number of terms in } d} \)
  - **idf**: inverse document frequency:
    - Document frequency \( df(t) \): number of documents in a collection that contains term \( t \)
    - \( N \): total number of documents in the collection
    - \( \text{Idf}(t) = \log (N / df(t)) \)

\[ \text{tf-idf}(t,d) = tf(t,d) \times \text{Idf}(t) \]
Bag-of-Words Model (BOW): Example

D1: The food is so cheap and the waiters are nice.
D2: The food was fantastic but the waiter non-existent.

stemming

{ the, food, is, so, cheap, and, waiter, are, nice, was, and, but, non, existent} { the, food, be, so, cheap, and, waiter, nice, and, but, non, existent}

lemmatization

hashing trick

{ 0:the, 1:food, 2:is, 3:so, 4:cheap, 5:and, 6:waiter, 7:are, 8:nice, 9:was, 10:and, 11:but, 12:non, 13:existent} { 0:the, 1:food, 2:be, 3:so, 4:cheap, 5:and, 6:waiter, 7:nice, 8:and, 9:but, 10:non, 11:existent}
Bag-of-Words Model (BOW): Example

{ 0: the, 1: food, 2: is, 3: so, 4: cheap, 5: and, 6: waiter, 7: are, 8: nice, 9: was, 10: fantastic, 11: but, 12: non, 13: existent}

D1: [1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]
D2: [1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1]

D1: [2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]
D2: [2, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1]

Presence/absence

D1: [1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]
D2: [1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1]

frequency

D1: [2, 1, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]
D2: [2, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1]

tf-idf

Tfidf(food,D1)= 1/10*log (2/2)=0
Tfidf(food,D2)= 1/9*log (2/2)=0
Tfidf(cheap,D1)=1/10*log(2/1)=0.030
Tfidf(be,D1)= 2/10*log (2/2)=0
Tfidf(be,D2)= 1/9*log (2/2)=0
Tfidf(waiter,D2)=1/9*log(2/1)=0.030
Text Classification: Evaluation

Goal: assess the performances of the classifiers on a given task

1. The corpus is divided in train + test mapped to gold annotations (reference)
   
   NB: Train and test sets **must be disjoined**
   
   ➔ Method: learn the model on the train corpus, apply (validate) the model on the development corpus, evaluate the obtained predictions against gold annotations

2. There is only a train corpus available
   
   ➔ Learn the model on the training set and assess its performances through K-fold **cross-validation**
   
   ➔ Divide the train corpus in, for example, 80% train, 20% test
Text Classification: K-fold cross-validation

**Definition:** A method for estimating the accuracy of a model by dividing the data into **K mutually exclusive subsets** (the “folds”) of approximately equal size.

Example of K=3 fold cross validation

- In practice, K=5 or K=10
- Results are based on the average results of the k-fold prediction
- Method used to tune the parameters of a model
Text Classification: Evaluation Metrics

True positive TP = correctly identified instances
False positive FP = incorrectly identified instances
True negative TN = correctly rejected instances
False negative FN = incorrectly rejected instances

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Spam</td>
<td>Spam</td>
</tr>
<tr>
<td>2</td>
<td>Spam</td>
<td>spam</td>
</tr>
<tr>
<td>3</td>
<td>Standard</td>
<td>Standard</td>
</tr>
<tr>
<td>4</td>
<td>Standard</td>
<td>Spam</td>
</tr>
<tr>
<td>5</td>
<td>Spam</td>
<td>Spam</td>
</tr>
<tr>
<td>6</td>
<td>Spam</td>
<td>standard</td>
</tr>
<tr>
<td>7</td>
<td>Standard</td>
<td>standard</td>
</tr>
<tr>
<td>8</td>
<td>Standard</td>
<td>spam</td>
</tr>
<tr>
<td>9</td>
<td>Standard</td>
<td>Standard</td>
</tr>
<tr>
<td>10</td>
<td>Spam</td>
<td>Standard</td>
</tr>
<tr>
<td>11</td>
<td>Spam</td>
<td>Spam</td>
</tr>
<tr>
<td>12</td>
<td>Spam</td>
<td>Standard</td>
</tr>
</tbody>
</table>

Confusion matrix

- Precision
- Recall
- F-measure (F1)
- Accuracy
Text Classification: Evaluation Metrics

<table>
<thead>
<tr>
<th>Predicted</th>
<th>True</th>
<th>Standard</th>
<th>Spam</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>spam</td>
<td>2</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

Precision = \( \frac{TP}{TP + FP} \)
- Standard class: \( \frac{3}{3+3} = 0.50 \)
- Spam class: \( \frac{4}{4+2} = 0.66 \)

Recall = \( \frac{TP}{TP + FN} \)
- Standard class: \( \frac{3}{3+2} = 0.60 \)
- Spam class: \( \frac{4}{4+3} = 0.57 \)

F-Measure = \( \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \)
- Standard class: \( \frac{2 \times 0.5 \times 0.6}{0.5 + 0.6} \)
- Spam class: \( \frac{2 \times 0.66 \times 0.57}{0.66 + 0.57} \)

Accuracy = \( \frac{TP + TN}{TP + TN + FP + FN} \)
- \( \frac{3+4}{3+4+3+2} = 0.58 \)
Classification Applications in NLP

• Assigning subject categories, topics, genres
• Spam detection
• Authorship identification
• User profiling
  • Age/gender identification
  • Personality traits detection
• Language Identification
• Sentiment analysis (subjectivity detection, opinion polarity, emotions, ...)
• Stance / fake news classification
• ...

Sentiment Analysis
Introduction
What is Sentiment Analysis (1)

• *Movie*: is this review positive or negative?
• *Products*: what do people think about the new iPhone?
• *Public sentiment*: how is consumer confidence? Is despair increasing?
• *Politics*: what do people think about this candidate or issue?
• *Prediction*: predict election outcomes or market trends from sentiment
What is Sentiment Analysis (2)

• Opinion mining or sentiment analysis
  • Computational study of opinions, sentiments, subjectivity, evaluations, attitudes, appraisal, affects, views, emotions, etc., expressed in text.
  • Reviews, blogs, microblogs, discussions, news, comments, feedback, or any other documents

• “Opinions” are key influencers of our behaviors.
  • Our beliefs and perceptions of reality are conditioned on how others see the world.
  • Whenever we need to make a decision, we often seek out the opinions of others
Sentiment Analysis is a difficult problem (1)

• Negation
  • Direct: “I don’t like my new Iphone”
  • Scope: “I don’t recommend buying this Iphone”
  • Indirect: “Perhaps it is a great phone, but I fail to see why”

• Coreference:
  • “We watch the movie and went to dinner; it was awful”

• Slang and writing errors (c.f normalization & POS tagging)
  • Shortform: nite (night), sayin (saying), u(you)
  • Acronyms: lol (laugh out loud), iirc (if I remember correctly).
  • Writing Errors: wouls(would), rediculous (ridiculous).
  • Punctuation Errors: im (I’m), dont (d’ont).
  • Slang: that was well mint (that was very good).
  • Repeated Letters: that was soooooo greeeat (that was so great)
  • Alphanumeric Words: 2night(tonight), str8(straight). 32
Sentiment Analysis is a Difficult Problem (2)

• Comparative
  • “Federer is better than Nadal”
    • Federer (+), Nadal (-)

• Domain Dependent Opinion
  • “The battery life is long” (+)
  • “The waiting time to enter at restaurant was too long” (-)

• Many more challenges
  • Irony / sarcasm
    • “Jack Reacher: Never go back” : probably shouldn't have
    • “I love waking up with migraines #not :'( “
  • Implicit sentiments
    • “She is still looking for another Oscar. Not here though.”
  • Opinion Spam (fake reviews)
Subjectivity

Sentence subjectivity:

• Objective sentences contain factual information

• Subjective sentences expresses personal opinions, beliefs, views, feelings, desires, speculations, suspicions or emotions (Wiebe, 2000; Riloff et al 2005).
  • A subjective sentence may contain a positive or negative opinion

• Most opinionated sentences are subjective but objective (factual) sentences may imply opinions too
  • “The machine stopped working in the second day”
  • “After taking the drug, there is no more pain”
What is an Opinion?

• ID: Abc123 on 5-1-2008: “I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, ...”

• One can look at this review/blog at the
  • document level, i.e., is this review + or -?
  • sentence level, i.e., is each sentence + or -?
  • entity and aspect level
Entity and Aspect Level

• ID: Abc123 on 5-1-2008: “I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, ...”

• What do we see?
  • Opinion targets: main entities of the world and their features/aspects
  • Sentiments: positive, negative, sometimes neutral
  • Opinion holders: persons who hold the opinions
  • Time: when opinions are expressed
Opinion Definition (Liu, Ch. in NLP handbook, 2010)

- An opinion is a quintuple \((e_j, a_{jk}, s_{ijkl}, h_i, t_l)\)

Where:
- \(e_j\) is a target entity
- \(a_{jk}\) is an aspect of the entity \(e_j\)
- \(s_{ijkl}\) is the sentiment value of the opinion from the opinion holder \(h_i\) on aspect \(a_{jk}\) of entity \(e_j\) at time \(t_l\)
  - \(s_{ijkl}\) is positive, negative or neutral (\(+, -, \sim\))
- \(h_i\) is an opinion holder
- \(t_l\) is the time when the opinion is expressed
Entity and Aspect Level

• ID: Abc123 on 5-1-2008: “I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, ...”

• In quintuples

(Target, Aspect,Polarity, Opinion Holder , Time)
(iPhone, GENERAL,+ , Abc123, 5-1-2008)
(iPhone, touch_screen,+ , Abc123, 5-1-2008)
(iPhone, voice_quality,+ , Abc123, 5-1-2008)
Structured the Unstructured

**Goal**: given an opinionated document:

- **Aspect Based Sentiment Analysis**:
  - Discover all quintuples \((e_j, a_{jk}, s_{ijkl}, h_i, t_i)\)
- Or, solve some simpler form of the problem
  - Classify the sentiment of the entire document (document level)
  - Classify the sentiment of each sentence of the document:
    - Subjectivity detection
    - Sentiment classification
Opinion Summary

• With a lot of opinions, a summary is necessary
  • A multi-document summarization task
• For factual texts, summarization: select the most important facts
  • 1 fact = any number of the same fact
• But for opinion documents, it is different because opinions have a quantitative side & have targets
  • 1 opinion ≠ a number of opinions
  • Aspect-based summary is more suitable
    • Quintuples form the basis for opinion summarization
**Aspect-Based Opinion Summary**

- **ID**: Abc123 on 5-1-2008: “I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, ...”

**Feature Based Summary of iPhone**:

- **Aspect 1**: Touch screen
  - **Positive**: 212
    - The touch screen was really cool
    - The touch screen was so easy to use and can do amazing things.
  - **Negative**: 6
    - The screen is easily scratched.
    - I have a lot of difficulty in removing finger marks from the touch screen.

- **Aspect 2**: Voice quality
  ...
Datasets for Sentiment Analysis

• Customer reviews about restaurants, hotel, camera, laptops, mobile phone, movies ....
  • From TripAdvisor, Amazon Reviews, Imdb, Rotten Tomatoes

• Twitter
  • Generic tweets
  • Domain-oriented tweets (e.g. politics)

• Blogs and forums: specific applications

• Annotated Datasets
  • Manually (crowdsourcing), e.g. for academic challenges
  • Distant supervision (silver datasets):
    • Star rating for reviews
    • Hashtags or emoticons for tweets
Sentiment Lexicons
Sentiment Lexicons

• Dictionaries of well-known sentiment words (Some available for R&D)
  • Words associated with their semantic class
  • Emotions: happiness, anger, fear
  • Polarities: positive, negative, neutral

• Manually Build

• Automatically Build
  • Using a few labeled examples
  • Using a few hand-built patterns (seeds)

• Many sentiment applications rely on lexicons to supply features to a model
Manually Created Lexicon

• The General Inquirer http://www.wjh.harvard.edu/~inquirer
  • Categories:
    • Positive (1915 words) and Negative (2291 words)
    • Strong vs Weak, Active vs Passive, Overstated versus Understated
    • Pleasure, Pain, Virtue, Vice, Motivation, Cognitive Orientation, etc.

• LIWC (Linguistic Inquiry and Word Count) http://www.liwc.net/
  • 2300 words, >70 classes
  • Affective Processes
    • negative emotion (bad, weird, hate, problem, tough)
    • positive emotion (love, nice, sweet)
  • Cognitive Processes
    • Tentative (maybe, perhaps, guess), Inhibition (block, constraint)
  • Pronouns, Negation (no, never), Quantifiers (few, many)
Manually Created Lexicon

• MPQA Subjectivity Cues Lexicon (GNU GPL)
  • 6885 words, annotated for intensity (strong, weak)
    • 2718 positive
    • 4912 negative

• Bing Liu’s Lexicon


  • [http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar](http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar)
  • [Bing Liu's Page on Opinion Mining](http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar)
  • 6786 words
    • 2006 positive
    • 4783 negative
Manually Created Lexicons

• NRC Word-Emotion Association Lexicon

• 8 basic emotions, in four opposing pairs:
  • joy–sadness / anger–fear / trust–disgust / anticipation–surprise

• 10 000 words annotated via Mechanical Turk

• Example:

<table>
<thead>
<tr>
<th>Amazingly</th>
<th>Emotion</th>
<th>Value</th>
<th>Amazingly</th>
<th>Emotion</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazingly</td>
<td>anger</td>
<td>0</td>
<td>amazingly</td>
<td>sadness</td>
<td>0</td>
</tr>
<tr>
<td>amazingly</td>
<td>anticipation</td>
<td>0</td>
<td>amazingly</td>
<td>surprise</td>
<td>1</td>
</tr>
<tr>
<td>amazingly</td>
<td>disgust</td>
<td>0</td>
<td>amazingly</td>
<td>trust</td>
<td>0</td>
</tr>
<tr>
<td>amazingly</td>
<td>fear</td>
<td>0</td>
<td>amazingly</td>
<td>negative</td>
<td>0</td>
</tr>
<tr>
<td>amazingly</td>
<td>joy</td>
<td>1</td>
<td>amazingly</td>
<td>positive</td>
<td>1</td>
</tr>
</tbody>
</table>
Automatically Generated Lexical Resources

• SentiWordNet: [http://sentiwordnet.isti.cnr.it/](http://sentiwordnet.isti.cnr.it/)
  Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010 SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. LREC-2010

  • Words annotated with degrees of positivity and negativity
    • `estimable(J,1] “deserving of respect or high regard”`
      `Pos .75 Neg 0 Obj .25`

• NRC Twitter Lexicons: [http://saifmohammad.com/WebPages/lexicons.htm](http://saifmohammad.com/WebPages/lexicons.htm)
  • Distant supervision with emotion hashtags
  • Weighted lexical ngrams with PMI
Overview of tasks and methods for sentiment analysis
Methods for Sentiment Analysis

- Machine Learning approaches
  - ML algorithms + linguistic features
- Lexicon based approaches
  - Sentiment lexicons (pre-compiled terms)
- Corpus-based
- Hybrid approaches: combine both
- Most frequent algorithms
  - Naïve Bayes
  - Maximum Entropy
  - Logistic Regression
  - SVM
- Recently: Deep Learning: RNN (LSTMs), CNN & word embeddings
Document Level Sentiment Classification

• Classify a whole opinion document (e.g., a review, a blog post, a tweet) based on the overall sentiment of the opinion

• Classes: Positive, negative (possibly neutral)
  • Neutral or no opinion is hard (often ignored)

• An example review
  “I bought an iPhone a few days ago. It is such a nice phone, although a little large. The touch screen is cool. The voice quality is clear too. I simply love it!”

• Classification: positive or negative?
  ➔ Perhaps the most widely studied problem
Example of Lexicon-Based methods: Turney Algorithm


Extract a *phrasal lexicon* from reviews

- Discover the polarity of *phrases*
- Positive association: « subtle nuance »
- Negative association: « very cavalier »

1. POS tagging + pattern extraction
2. Estimate polarity of each phrase
3. Rate a review by the average polarity of its phrases
## Pattern Extraction: two-word phrases with adjectives

<table>
<thead>
<tr>
<th>First Word</th>
<th>Second Word</th>
<th>Third Word (not extracted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JJ</td>
<td>NN or NNS</td>
<td>anything</td>
</tr>
<tr>
<td>RB, RBR, RBS</td>
<td>JJ</td>
<td>Not NN nor NNS</td>
</tr>
<tr>
<td>JJ</td>
<td>JJ</td>
<td>Not NN or NNS</td>
</tr>
<tr>
<td>NN or NNS</td>
<td>JJ</td>
<td>Nor NN nor NNS</td>
</tr>
<tr>
<td>RB, RBR, or RBS</td>
<td>VB, VBD, VBN, VBG</td>
<td>anything</td>
</tr>
</tbody>
</table>

POS-based Phrases extracted from Reviews
Estimate Polarity of a Phrase

• Positive phrases co-occur more with “excellent”, negative phrases co-occur more with “poor”

• Use Pointwise **mutual information (PMI)** between two words
  
  = How much more do two words co-occur than if they were independent

\[
\text{PMI}(\text{word}_1, \text{word}_2) = \log_2 \frac{P(\text{word}_1, \text{word}_2)}{P(\text{word}_1)P(\text{word}_2)}
\]

Polarity(phrase) = PMI(phrase,"excellent") - PMI(phrase,"poor")

Query search engine (Altavista)

  P(word) estimated by  hits(word)/N

  P(word\_1,word\_2) by  hits(word\_1 NEAR word\_2)/N
# Examples of phrases and polarities

<table>
<thead>
<tr>
<th>Phrase</th>
<th>POS tags</th>
<th>Polarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>online service</td>
<td>JJ NN</td>
<td>2.8</td>
</tr>
<tr>
<td>online experience</td>
<td>JJ NN</td>
<td>2.3</td>
</tr>
<tr>
<td>direct deposit</td>
<td>JJ NN</td>
<td>1.3</td>
</tr>
<tr>
<td>local branch</td>
<td>JJ NN</td>
<td>0.42</td>
</tr>
<tr>
<td>low fees</td>
<td>JJ NNS</td>
<td>0.33</td>
</tr>
<tr>
<td>true service</td>
<td>JJ NN</td>
<td>-0.73</td>
</tr>
<tr>
<td>other bank</td>
<td>JJ NN</td>
<td>-0.85</td>
</tr>
<tr>
<td>inconveniently</td>
<td>JJ NN</td>
<td>-1.5</td>
</tr>
<tr>
<td>located</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.32</td>
</tr>
</tbody>
</table>

**Positive review**

<table>
<thead>
<tr>
<th>Phrase</th>
<th>POS tags</th>
<th>Polarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>direct deposits</td>
<td>JJ NNS</td>
<td>5.8</td>
</tr>
<tr>
<td>online web</td>
<td>JJ NN</td>
<td>1.9</td>
</tr>
<tr>
<td>very handy</td>
<td>RB JJ</td>
<td>1.4</td>
</tr>
<tr>
<td>virtual monopoly</td>
<td>JJ NN</td>
<td>-2.0</td>
</tr>
<tr>
<td>lesser evil</td>
<td>RBR JJ</td>
<td>-2.3</td>
</tr>
<tr>
<td>other problems</td>
<td>JJ NNS</td>
<td>-2.8</td>
</tr>
<tr>
<td>low funds</td>
<td>JJ NNS</td>
<td>-6.8</td>
</tr>
<tr>
<td>unethical practices</td>
<td>JJ NNS</td>
<td>-8.5</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>-1.2</td>
</tr>
</tbody>
</table>

**Negative review**
Evaluation of Turney Algorithm on Review Classification

• 410 reviews from Epinions (customer reviews)
  • 170 (41%) negative
  • 240 (59%) positive

• Majority class baseline: 59%

• Turney algorithm: compute the average polarity for each review and select the label with the highest score: 74%

• Use phrases rather than words

• Advantage: Learns domain-specific information (corpus-based method)
Supervised Classifiers for Document-Level Sentiment


• Baseline algorithm for classifying documents (movie reviews) by overall sentiment (positive or negative)

• Examine the effectiveness of different classification algorithms to the movie review domain

• Sentiment classification more challenging than topic-based classification.
  • “How could anyone sit through this movie?”
    • no word that is obviously negative
  • Sentiment seems to require more understanding than the usual topic-based classification.
Data: Movie Reviews

• Data source: The IMDB archive

• Selected reviews where author rating was expressed with stars or numerical value
  • Automatically converted to one of three categories: positive, negative, or neutral.
  • Impose a limit of fewer that 20 reviews per author per sentiment category.
    • 1301 positive reviews
    • 752 negative reviews
    • 144 reviewers

• Finally random selection of 700 positive reviews and 700 negative reviews
Naïve Approach

Idea: people tend to use certain words to express strong sentiments
• Produce such list (human annotation)
• Rely on it to classify text (counting occurrences)

<table>
<thead>
<tr>
<th>Proposed word lists</th>
<th>Accuracy</th>
<th>Ties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human 1</td>
<td>58%</td>
<td>75%</td>
</tr>
<tr>
<td>Positive: dazzling, brilliant, phenomenal, excellent, fantastic negative: suck, terrible, awful, unwatchable, hideous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human 2</td>
<td>64%</td>
<td>39%</td>
</tr>
<tr>
<td>Positive: gripping, mesmerizing, riveting, spectacular, cool, awesome, thrilling, badass, excellent, moving, exciting negative: bad, cliched, sucks, boring, stupid, slow</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tie rate: percentage of documents where the 2 sentiments are rated equally

Figure 1: Baseline results for human word lists. Data: 700 positive and 700 negative reviews.

<table>
<thead>
<tr>
<th>Proposed word lists</th>
<th>Accuracy</th>
<th>Ties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human 3 + stats</td>
<td>69%</td>
<td>16%</td>
</tr>
<tr>
<td>Positive: love, wonderful, best, great, superb, still, beautiful negative: bad, worst, stupid, waste, boring, ?, !</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Results for baseline using introspection and simple statistics of the data (including test data).
Machine Learning Approach

• Bag of features
• Supervised classification methods
  • Naïve Bayes
  • Maximum Entropy
  • SVM
Machine Learning Approach

• Features
  • Unigrams & bigrams
  • POS
  • Adjectives
  • Position of the unigram
    • The word appear in the first quarter, the last quarter or the middle of the movie review

• Scoring
  • Frequency or presence
## Machine Learning Approach: Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Features</th>
<th># of features</th>
<th>frequency or presence?</th>
<th>NB</th>
<th>ME</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>unigrams</td>
<td>16165</td>
<td>freq.</td>
<td><strong>78.7</strong></td>
<td>N/A</td>
<td>72.8</td>
</tr>
<tr>
<td>2</td>
<td>unigrams</td>
<td></td>
<td>pres.</td>
<td>81.0</td>
<td>80.4</td>
<td><strong>82.9</strong></td>
</tr>
<tr>
<td>3</td>
<td>unigrams+bigrams</td>
<td>32330</td>
<td>pres.</td>
<td>80.6</td>
<td>80.8</td>
<td><strong>82.7</strong></td>
</tr>
<tr>
<td>4</td>
<td>bigrams</td>
<td>16165</td>
<td>pres.</td>
<td><strong>77.3</strong></td>
<td><strong>77.4</strong></td>
<td>77.1</td>
</tr>
<tr>
<td>5</td>
<td>unigrams+POS</td>
<td>16695</td>
<td>pres.</td>
<td>81.5</td>
<td>80.4</td>
<td><strong>81.9</strong></td>
</tr>
<tr>
<td>6</td>
<td>adjectives</td>
<td>2633</td>
<td>pres.</td>
<td>77.0</td>
<td><strong>77.7</strong></td>
<td>75.1</td>
</tr>
<tr>
<td>7</td>
<td>top 2633 unigrams</td>
<td>2633</td>
<td>pres.</td>
<td>80.3</td>
<td>81.0</td>
<td><strong>81.4</strong></td>
</tr>
<tr>
<td>8</td>
<td>unigrams+position</td>
<td>22430</td>
<td>pres.</td>
<td>81.0</td>
<td>80.1</td>
<td><strong>81.6</strong></td>
</tr>
</tbody>
</table>

Figure 3: Average three-fold cross-validation accuracies, in percent. Boldface: best performance for a given setting (row). Recall that our baseline results ranged from 50% to 69%.
Sentiment Analysis in Twitter

• Microblogging: very popular communication tool
• Twitter: most popular, 240+ million active users, 500 million tweets/day
• Short message text 140 characters
  • Tweets are short, highly unstructured, non grammatical and ambiguous
  • Out of vocabulary words (lexical creativity)
  • Elongated words: niiiiiiiiiiice, saaaaaaad
  • Extensive usage of acronyms like asap, lol, afaik
  • Hashtags, emoticons
• Personal views on various subjects and current events
• A lot of works in sentiment analysis focus on twitter data
Tasks

• Phrase level sentiment analysis
  • Given a tweet containing a marked topic, determine the polarity about this topic
  • Detecting the topic can also be part of the task

• Sentence level sentiment analysis
  • Given a tweet decide whether it is positive negative or neutral
Standard Architecture

Tweet Training set → Tokenization → Preprocessing/Filtering → Feature extraction → Feature vectors 1 per tweet → Classification Algorithm

Tweet Test set → Tokenization → Preprocessing/Filtering → Feature extraction → Feature vectors 1 per tweet → Predictive model → Predicted Labels

Labels
Examples of Preprocessing of Tweets

• Remove all URLs (e.g. www.xyz.com), targets (@username)
• Remove hashtags (#topic)
• Correct the spellings; handle sequence of repeated characters (niiiiiiice)
• Replace all the emoticons with their sentiment (if known)
• Remove all punctuations, symbols, number
• Remove stop words
• Expand acronyms (acronym dictionary)
• Remove Non-English Tweets (e.g. with language identifier)
• POS tagging
Sample Features for Tweets classification

• Word ngrams: presence of contiguous sequences of 1, 2, 3 and 4 tokens; noncontiguous ngrams
• POS: the number of occurrence of each POS TAG
• Sentiment Lexica each word annotated with tonality score (+1,...0,...-1)
• Negation: number of negated contexts
• Punctuation: the number of contiguous sequences of exclamation marks, question marks and both
• Emoticons: presence or absence, last token is a positive or negative emoticon
• Hashtags: number, presence
• Elongated words: number, presence
SemEval Challenges on Tweet Classification

• Challenge since 2013
  • Main task: given a tweet, decide whether it is positive, negative or neutral

• Evolutions:
  • From tweet level opinion classification to more fine grained analysis
    • Sentiment toward a topic classification
    • Quantification of the distribution of sentiment towards a topic across a number of tweets
  • New languages, e.g. Arabic

• Data:
  • Collected via twitter API on trendy topics (Trump, gun control, …)
  • Annotated via crowdsourcing (~50000 tweets for train, ~12000 for test)

• 2017 edition: 48 teams participated – Very popular task
[Mohammad et al 2013] NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. SemEval 2013, Atlanta, USA

• Tokenization and POS tagging with CMU tools (cf. POS Tagger [Gimpel 2011])

• Features:
  • Word ngrams (1, 2, 3, 4)
  • Character ngrams
    • contiguous sequences of 3, 4, 5 characters
  • All-caps (number of words)
  • POS (number of occurrences)
  • Number of hashtags
  • Sentiment Lexicon features
  • Polarity of emoticons
  • Number of elongated words
  • Negation: number of negated contexts

• Classification with Support Vector Machines

<table>
<thead>
<tr>
<th></th>
<th>Classifier</th>
<th>Tweets</th>
<th>SMS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training set:</strong></td>
<td>Majority</td>
<td>26.94</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>SVM-all</td>
<td>67.20</td>
<td>-</td>
</tr>
<tr>
<td><strong>Development set:</strong></td>
<td>Majority</td>
<td>26.85</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>SVM-all</td>
<td>68.72</td>
<td>-</td>
</tr>
<tr>
<td><strong>Test set:</strong></td>
<td>Majority</td>
<td>29.19</td>
<td>19.03</td>
</tr>
<tr>
<td></td>
<td>SVM-unigrams</td>
<td>39.61</td>
<td>39.29</td>
</tr>
<tr>
<td></td>
<td>SVM-all</td>
<td>69.02</td>
<td>68.46</td>
</tr>
</tbody>
</table>

F1 scores
SemEval2017 Challenge on Sentiment Tweets

- Increase of method using deep learning (CNN, LSTM) ~20 teams
- SVM + Neural networks or SVM + words embeddings
- Common software used included Python (sklearn, numpy libraries), Java, Tensorflow, Weka, NLTK, Keras, Theano, and StanfordCoreNLP.
- Additional resources
  - Sentiment140 (Stanford) corpus for learning word embeddings
  - Additional tweets collected with twitter API
- Best systems on polarity classification
  - DataStories used deep LSTM networks with an attention mechanism (accuracy 65.8)
  - BBtwtrr used ensemble of LSTMs and CNNs with multiple convolution operations (accuracy 65.1)
Word embeddings and sentiment classification

Problem for sentiment analysis tasks

- Existing word embedding learning algorithms only use the contexts of words but ignore the *sentiment* of texts.
- Words with similar contexts but opposite sentiment polarity, such as *good* and *bad*, are mapped to neighboring word vectors, because of similar usage and grammatical rules.

**Sentiment-Specific Word Embeddings (SSWE)**

Tang et al., Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification, Proceedings of the 52nd ACL meeting, Baltimore 2014
Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification

• Extension of Collobert & Weston (C&W) Word Embedding learning method
• Neural networks with different strategies to **effectively incorporate the supervision from sentiment polarity in their loss function**
• Learn from a massive corpus of tweets with emoticons as distant supervised corpora
• Apply (and evaluate) SSWE as feature for twitter sentiment classification (c.f. SemEval2013 tweet classification)
Collobert & Weston (C&W)

- Given an ngram, C&W replaces the center word with a random word to derive a corrupted ngram.
- Training objective: original ngram is expected to obtain a higher language model score than the corrupted ngram by a margin of 1.

**Loss Function**

\[
\max(0, 1 - f_{cw}(t) + f_{cw}(t^r))
\]

- Original Ngram
- Corrupted Ngram
Model SSWE

- Predict the sentiment polarity based on each ngram
- Training objective: if the sentiment polarity of a tweet is positive, the predicted positive score is expected to be larger than the predicted negative score, and the exact reverse if the tweet has negative polarity

Loss Function
\[
\max(0, 1 - \delta_s(t) f^r_0(t) + \delta_s(t) f^r_1(t))
\]

Indicator Function
\[
\delta_s(t) = \begin{cases} 
1 & \text{if } f^g(t) = [1, 0] \\
-1 & \text{if } f^g(t) = [0, 1]
\end{cases}
\]
Combination of C&W and SSWE

- Use both the syntactic contexts of words and the sentiment polarity of sentences to learn the sentiment-specific word embedding.
- Given an original (or corrupted) ngram and the sentiment polarity of a sentence as input, predict a two-dimensional vector $(f_0, f_1)$ for language model score and sentiment score.

**Loss Function**

$$\alpha \cdot \text{loss}_{cw}(t, t^r) + (1 - \alpha) \cdot \text{loss}_{us}(t, t^r)$$

**Sentiment Loss**

$$\max(0, 1 - \delta_s(t) f_{1u}^u(t) + \delta_s(t) f_{1u}^u(t^r))$$
Embedding Training

• Data
  • Tweets contains positive/negative emoticons
    
    | Positive | :) | : ) | :-) | :D | =) |
    |----------|----|------|------|----|-----|
    | Negative | :( | : ( | :-( |    |     |

  • 5M positive, 5M negative tweets from April, 2013

• Back-propagation + AdaGrad [Duchi 2011]
  • Embedding length = 50
  • Window size = 3
  • Learning rate = 0.1
Twitter Sentiment Classification

- Evaluation of SSWE through a supervised learning framework for sentiment classification
- Sentiment Classifier trained with LibLinear
- Dataset from SemEval 2013

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
<th>Neutral</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>2,642</td>
<td>994</td>
<td>3,436</td>
<td>7,072</td>
</tr>
<tr>
<td>Dev</td>
<td>408</td>
<td>219</td>
<td>493</td>
<td>1,120</td>
</tr>
<tr>
<td>Test</td>
<td>1,570</td>
<td>601</td>
<td>1,639</td>
<td>3,810</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the SemEval 2013 Twitter sentiment classification dataset.
## Results (1)

- Comparison with different twitter sentiment classification algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistSuper + unigram</td>
<td>61.74</td>
</tr>
<tr>
<td>DistSuper + uni/bi/tri-gram</td>
<td>63.84</td>
</tr>
<tr>
<td>SVM + unigram</td>
<td>74.50</td>
</tr>
<tr>
<td>SVM + uni/bi/tri-gram</td>
<td>75.06</td>
</tr>
<tr>
<td>NBSVM</td>
<td>75.28</td>
</tr>
<tr>
<td>RAE</td>
<td>75.12</td>
</tr>
<tr>
<td><strong>NRC (Top System in SemEval)</strong></td>
<td>84.73</td>
</tr>
<tr>
<td>NRC - ngram</td>
<td>84.17</td>
</tr>
<tr>
<td>$SSWE_u$</td>
<td>84.98</td>
</tr>
<tr>
<td>$SSWE_u+NRC$</td>
<td>86.58</td>
</tr>
<tr>
<td>$SSWE_u+NRC$-ngram</td>
<td>86.48</td>
</tr>
</tbody>
</table>
Results(2)

Comparison with different word Embeddings
Aspect Based Sentiment Analysis (ABSA)
Aspect-Based Sentiment Analysis (ABSA)

- **ABS A:** Fine-grained opinion annotation
  - Determine *sentiments* about different *aspects* of *entities* (e.g. movies, restaurants, cell phones,...)
  - *Aspects are features* of an entity (*service*, *food* in a restaurant; *screen*, *battery* of a cell phone,...)
  - Human annotation of data is very costly (= small annotated datasets)

<table>
<thead>
<tr>
<th>Term</th>
<th>Aspect</th>
<th>Polarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quinoa salad, mussels</td>
<td>Food Quality</td>
<td>positive</td>
</tr>
<tr>
<td>Quinoa salad, mussels</td>
<td>Food Quantity</td>
<td>negative</td>
</tr>
<tr>
<td>Terrace, sea view</td>
<td>Location</td>
<td>positive</td>
</tr>
<tr>
<td>waiter</td>
<td>Service</td>
<td>negative</td>
</tr>
</tbody>
</table>
An Example: ABSA @ SemEval2016

[Brun, Perez, Roux 2016]: “XRCE at SemEval-2016 Task 5: Feedbacked Ensemble Modeling on Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis”, SemEval@NAACL-HLT 2016, San Diego, CA, USA, 2016},

• Task description
• Presentation of the system
• Results
• Other Systems
SemEval 2016 Task 5 ABSA Description

• Mining opinions from text about specific entities and their aspects (c.f. Bing Liu’s definition)

• Broad variety of domains and languages

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Dutch</th>
<th>French</th>
<th>Russian</th>
<th>Turkish</th>
<th>Arabic</th>
<th>Chinese</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restaurants</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Hotels</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✔️</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Laptops</td>
<td>✔️</td>
<td>✔️</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✔️</td>
<td>-</td>
</tr>
<tr>
<td>Telecom</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✔️</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Example of Annotation

“Thereir sake list was extensive, but we were looking for Purple Haze, which wasn't listed but made for us upon request!”

<Opinions>
<Opinion target="sake list" category="DRINKS#STYLE_OPTIONS" polarity="positive" from="6" to="15"/>
<Opinion target="NULL" category="SERVICE#GENERAL" polarity="positive" from="0" to="0"/>
</Opinions>

Sentence level ABSA: 3 subtasks
1. Aspect category detection (Topic labelling with semantic categories)
2. Opinion Target Expression (OTE = detecting opinionated terms of the domain)
3. Sentiment polarity (positive, negative, neutral)

Training corpus:
1733 annotated sentences (336 reviews) for French - 2000 annotated sentences (351 reviews) for English
Challenges

Fine-grained aspect categories (ontology of 12 classes)

food#quality, food#style_options, food#prices drinks#quality, drinks#style_options, drinks#prices, restaurant#general, restaurant#misc, restaurant#prices, service#general, ambience#general, location#general

Mixture of:

• Term level annotation
• Sentence level annotation, when an aspect category is expressed without a term (“NULL” = implicit opinion target)

Multiclass annotation: several categories and polarities possible for a given OTE/sentence

Small Dataset
ABSA: General Architecture of our System

Review Sentences

POS Tagging → Syntactic Parsing → Partial Semantic Parsing

Domain Lexicon (Aspects, Polarity)

Weakly-supervised lexical acquisition

Term Detection

CRF → Aspects Identification LogReg → Polarity Identification MaxEnt

Representation

BoW / N grams

POS Tags

Syntactic Relations

Semantic Relations

Word Embeddings

Word Polarity and Aspects

FOOD_QUALITY → POSITIVE
DRINKS_PRICE → NEGATIVE
AMBIENCE → POSITIVE
SERVICE → NEGATIVE

Labs
Algorithms

• CRF for opinionated term detection (sequence labelling)
• Logistic regression for aspect and MaxEnt for polarity classification
  • Combination of different models and synthetize the results in one score
• Feature extraction using our in-house opinion grammars (parsing step)
  • For term detection
  • For term classification into aspect and polarities
  • For sentence classification into aspect and polarities (*NULL cases*)
• Feature selection: Large amount of lexicalized features
  • Feature selection by thresholding to eliminate rare feature (nb_occ>=5)
  • Feature selection by subsampling:
    • Cross-validation through subsampling of the overall feature space to select the best features
Linguistic Feature Extraction

• Based on (in house) opinion detection grammars for French and English
  • Lexical semantics about the domain (food, service, etc ....)
  • Contextual polarities

• Features:
  • bow, bigrams, trigrams
  • Lexical semantics
  • Parsing Dependencies : SUBJ, OBJ, ATTRIB, NMOD, VMOD, SENTIMENT
  • Term Centric features: dependencies and n-gram involving a term
  • Sentence-based features: all dependencies & n-grams
  • Features are de-lexicalized for polarity classification
    SUBJ(like, term_food), SENTIMENT_POS(term_service, nice)
Annotation Process – Prediction Strategy

1. **OTE (Terms) detection** with CRF

2. Classification of **terms** into aspect categories (multiclass: highest probability + others above a threshold computed at cross-validation)

3. Classification of **sentences** into aspect categories (case “NULL”, multiclass: probabilities above a certain threshold computed at cross-validation)

4. Classification of **terms and sentences** detected previously into aspect polarities, with best probability (even if in some rare cases there are different polarities for the same <aspect term, aspect category>)
Evaluation

**Phase A:** OTE detection, Aspect categories classification

**Phase B:** Ground truth of Phase A distributed to participants to annotate polarities

800 sentences for testing on English and French

4 Evaluation measures:

- F-score for aspect category detection (S1)
- F-score for OTE detection (S2)
- F-score for tuples <aspect term, aspect cat> (S12)
- Accuracy for polarity detection (S3)
## Results of our System

### Phase A

<table>
<thead>
<tr>
<th>Language</th>
<th>Aspect Categories S1</th>
<th>OTE (aspect terms) S2</th>
<th>Tuples &lt;OTE,cat&gt; S12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>Rank</td>
<td>Basel.</td>
</tr>
<tr>
<td>English</td>
<td>68.70</td>
<td>7/30</td>
<td>59.92</td>
</tr>
<tr>
<td>French</td>
<td><strong>61.21</strong></td>
<td>1/5</td>
<td>52.61</td>
</tr>
</tbody>
</table>

### Phase B

<table>
<thead>
<tr>
<th>Language</th>
<th>Aspect Polarities S3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>English</td>
<td><strong>88.13</strong></td>
</tr>
<tr>
<td>French</td>
<td><strong>78.82</strong></td>
</tr>
</tbody>
</table>

~ 32 teams participated to this challenge
IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis
[Kumar et al. 2016] SemEval@NAACL2016

- Polarity detection using Support Vector Machine (SVM)
- Preprocessing: Tokenize, parse, extract lemma, POS, NE information with Stanford parser and parsers based on UD; Remove stop words
- Feature extraction:
  - Lexical features based on distributional thesaurus (expansion of existing lexicons)
  - Unigrams and bigrams as binary features

<table>
<thead>
<tr>
<th>Token</th>
<th>DT Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>bad, excellent, decent, great</td>
</tr>
<tr>
<td>powerful</td>
<td>potential, influential, strong, sophisticated</td>
</tr>
<tr>
<td>small</td>
<td>tiny, large, sized, huge, sizable</td>
</tr>
<tr>
<td>efficient</td>
<td>reliable, effective, energy-efficient, flexible</td>
</tr>
</tbody>
</table>

Examples of DT expansions

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Polarity Classification: Acc. (Rank / Entries)</th>
</tr>
</thead>
<tbody>
<tr>
<td>English Restaurants</td>
<td>86.70 (2 / 29)</td>
</tr>
<tr>
<td>Dutch Restaurants</td>
<td>76.90 (2 / 4)</td>
</tr>
<tr>
<td>Spanish Restaurants</td>
<td>83.50 (1 / 5)</td>
</tr>
<tr>
<td>French Restaurants</td>
<td>72.20 (5 / 6)</td>
</tr>
<tr>
<td>Russian Restaurants</td>
<td>73.60 (3 / 6)</td>
</tr>
<tr>
<td>Turkish Restaurants</td>
<td>84.20 (1 / 3)</td>
</tr>
</tbody>
</table>

Results on polarity
A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis

Pure deep learning (LSTMs) to capture sentences and review context for aspect-based polarity classification

- Sentence level BiLSTM to capture sentence context
- Review level bidirectional LSTM to capture review context
- Aspect: feed the aspect representation together with the sentence representation into the review level BiLSTM
More prior knowledge $\rightarrow$ Less training data

<table>
<thead>
<tr>
<th>Participant (rank)</th>
<th>SemEval Baseline</th>
<th>Naver Baseline</th>
<th>Naver (1)</th>
<th>IIT-T (2)</th>
<th>(Khalil and El-Beltagy, 2016) (3)</th>
<th>(Ruder et al., 2016) (7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>BoW + SVM</td>
<td>syntactic features only + LR</td>
<td>Syntactic &amp; Semantic features + LR</td>
<td>Distributional semantics + SVM</td>
<td>CNNs + manual features + word embeddings</td>
<td>CNNs + word embeddings</td>
</tr>
<tr>
<td>Accuracy trained on 100% dataset (2000 sent.)</td>
<td>76.48</td>
<td>77.42</td>
<td>88.13</td>
<td>86.72</td>
<td>85.45</td>
<td>82.07</td>
</tr>
<tr>
<td>Accuracy trained on 20% dataset (400 sent.)</td>
<td>64.28</td>
<td>73.32</td>
<td>85.68</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- SemEval-2016 ABSA for restaurants, **polarity detection**, English
- Naver system: when trained on 20% of data only: rank=2\textsuperscript{nd}
  - Better than a CNN model trained on 100% of data (2000 sentences) + external dataset

$\rightarrow$ Linguistic prior knowledge significantly decreases the need for manual data annotation
Fake News & Stance Detection
“Fake News” are not new...

New York Sun – 1835
“Great Astronomical Discovery Lately made”

Weekly Word News – 1992
... but become viral nowadays!

Pope Francis Shocks World, Endorses Donald Trump for President, Releases Statement

**TOPICS:**  Pope Francis Endorses Donald Trump
Social Media: Main Driver for “Fake News”

- 10% of readers of top news come via social media
- 40% of readers of “fake news” come via social media
Fake News: a definition

• Fake news is where individuals or organizations intentionally publish hoaxes, propaganda and other misinformation and present it as factual (through blogs, social media posts, fake online media releases).

• It does not include news satire sites (e.g. The Onion or Le Gorafi) as they are not presenting their content as legitimate factual news. Their intention is satire rather than misinformation.

• It also does not include articles that are written from the perspective of a particular opinion or editorial standpoint, provided the information included is factually correct.
Automatic Fake News Detection

• Highly challenging problem

• Data used: satirical news source (The Onion), fact-checking websites such as Polifact and Snopes

• Drawbacks:
  • Satirical sources make content underlying confounding factors like humor and irony
  • Fact checking web sites are restricted to some domains such as politics and require human expertise to verify new claims
Automatic Fake New Detection

• Linguistic approach: deceptive content detection
  • Based on text properties: writing style & content (POS, punctuation, ...)
  • Best classification performance: feature set that includes punctuation, grammar, and psychological words (c.f. LIWC)

• Fact checking: assess truthfulness of news article claims
  • Querying knowledge base like Dbpedia and use the result to test whether different sources contain information confirming the claim (similarity)
  • Use social network activity, e.g. tweets voicing skepticism

• Stance Detection
  • Facts are highly complex and difficult to check
  • Instead: comparing how reputable sources feel about a claim
Stance Detection

• Automatically determine from a text whether the author is in favor / against / neutral with regards to a target topic (usually controversial).

• Examples:
  • Target: feminist movement
    Post: “Job should always go to best candidate, regardless of gender. Gender shouldn’t even matter anymore, it’s 2015! #PaulHenry”
  • Target: Bernie Sanders
    Post: “Be prepared - if we continue the policies of the liberal left, we will be #Greece”

• Applied to microblogs like tweeter, comments on news headlines or political debates
Key Challenges in Stance Detection

• The target topic of interest may not be the same as the target of the opinion:
  • Target: Donald Trump
    Post: “Jeb Bush is the only sane candidate for 2016”

• Main difference with (aspect-based) sentiment analysis: one may express favor towards a target by using negative language and vice versa
  • Target: legalization of abortion
    Post: “So not only are antichoice strongly against pregnant people’s human rights, they’re also homophobic. Shocker.”

• Other difference: stance can be expressed without using sentiment-bearing words
  • Target: Mammography does not reduce breast cancer deaths
    Post: “Studies show this over and over. The value of mammograms has not been proved.”
SemEval-2016 Stance Detection Challenge

- Target: *Hillary Clinton* (context: Party presidential primaries for Democratic and Republican parties in US)
- The tweet expresses a **positive opinion** toward an adversary of the target (*Sanders*)
- One can infer that the tweeter expresses a **negative stance** toward Hillary Clinton
- The tweet does not contain any explicit clue to find the target
- **In many cases the stance must be inferred**
- Dataset: [http://www.saifmohammad.com/WebPages/StanceDataset.htm](http://www.saifmohammad.com/WebPages/StanceDataset.htm)
  - 5 targets: ‘Atheism’, ‘Climate Change is a Real Concern’, ‘Feminist Movement’, ‘Hillary Clinton’, and ‘Legalization of Abortion’
  - 4870 English tweets annotated with stance
SemEval-2016 Stance Detection Challenge

- Majority class: strong baseline
- Unigram and ngram features bring substantial improvements
- Training separates classifiers is a good idea
- Best system: MITRE:
  - 2 RNNs classifiers: the first was trained to predict task-relevant hashtags on a very large unlabeled Twitter corpus.
  - This network was used to initialize a second RNN classifier for the task.
- All systems under baseline!
Summary
NLP for Social Media Analytics

• Social Media Data
  • Big data from user generated content
  • Value: mine human behavior, opinions, profiles....
  • Applications : health, business, security
  • Many opportunities for NLP: great interest in both academic and industry

• Many challenges for NLP
  • Noisy input
  • Natural complexity of language
    • Implicit opinions, emotions, subjectivity
    • Figurative language (irony, sarcasm)

• Current trends
  • NLP tools for twitter
  • Aspect Based Sentiment & summarization
  • Fake news, stance detection & argument mining
Evaluation:
Implement a (simplified) Aspect Based Sentiment Analysis Classifier

http://nlpcourse.europe.naverlabs.com/#exercise2
Backup slides
Inter annotator agreement: Cohen’s kappa $\kappa$

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

$p_o$ is the relative observed agreement among raters (~accuracy)

$p_e$ is the hypothetical probability of chance agreement

$K=1$ if complete agreement

$p_o = \frac{a + d}{a + b + c + d} = \frac{20 + 15}{50} = 0.7$

<table>
<thead>
<tr>
<th></th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>15</td>
</tr>
</tbody>
</table>

- Reader A said "Yes" to 25 applicants and "No" to 25 applicants. Thus reader A said "Yes" 50% of the time.
- Reader B said "Yes" to 30 applicants and "No" to 20 applicants. Thus reader B said "Yes" 60% of the time.

$$p_{Yes} = \frac{a + b}{a + b + c + d} \cdot \frac{a + c}{a + b + c + d} = 0.5 \times 0.6 = 0.3$$

$$p_{No} = \frac{c + d}{a + b + c + d} \cdot \frac{b + d}{a + b + c + d} = 0.5 \times 0.4 = 0.2$$

$$p_e = p_{Yes} + p_{No} = 0.3 + 0.2 = 0.5$$

$$\kappa = \frac{p_o - p_e}{1 - p_e} = \frac{0.7 - 0.5}{1 - 0.5} = 0.4$$
Exercise

• Create a feature vector representation for the corpus:
  • D1: “I am feeling very happy today”
  • D2: “I was feeling very sorry yesterday”
  • D3: “I was sorry yesterday but happy today”

1. Bow: stemming + presence
2. Bow + bigram: lemmatization + occurrences
Solution

D1: “I am feeling very happy today”
D2: “I was feeling very sorry yesterday”
D3: “I was sorry yesterday but happy today”

1. V:{0:l,1:am,2:feel,3:very,4:happy,5:today,6:was,7:sorry 8:yesterday,9:but}
   D1:[1, 1, 1, 1, 1, 1, 0, 0, 0, 0 ]
   D2:[1, 0, 1, 1, 0, 0, 0, 1, 1, 1]
   D3:[1, 0, 0, 0, 1, 1 ,1 , 1, 1, 1]

2. V:{0:l, 1:be, 2:feel, 3:very, 4:happy, 5:today, 6:sorry 7:yesterday, 8:but, 9: I be, 10:be feel, 11:feel very, 12:very happy, 13: happy today, 13:very sorry,14:sorry yesterday,15:be sorry,16:sorry yesterday,17:yesterday but, 18: but happy}
Exercise

• Accuracy
• precision, recall, F1

<table>
<thead>
<tr>
<th>Predicted</th>
<th>gold</th>
<th>cat</th>
<th>dog</th>
<th>rabbit</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>rabbit</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
Exercise

• Accuracy
  \[
  \frac{4+5+4}{4+5+4+1+2+2+1+2+0} = 0.684
  \]

• precision, recall, F1

<table>
<thead>
<tr>
<th></th>
<th>gold</th>
<th>cat</th>
<th>Dog</th>
<th>rabbit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cat</strong></td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>dog</strong></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td><strong>rabbit</strong></td>
<td>2</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

P-cat = 4/7
R-cat = 4/8
F1-cat = 0.53

P-dog = 5/8
R-dog = 5/6
F1-dog = 0.71

P-rabbit = 4/6
R-rabbit = 4/7
F1-rabbit = 0.61
Reminder: Word Embeddings

- Language modeling and feature learning techniques in NLP where words or phrases from the vocabulary are mapped to vectors of real numbers

- From a sparse representation to a dense representation
- Embeddings created as by product versus explicit models
- Low-dimensional and dense
- Semantic relationships between words are reflected in the distance and direction of the vectors.
Methods for Sentiment Analysis

Sentiment analysis

- Machine Learning approach
  - supervised
    - Decision tree classifiers
      - SVM
    - Linear classifiers
    - Rule-based classifiers
    - Probabilistic classifiers
      - Naive Bayes
      - Maximum Entropy
  - unsupervised
    - Deep Learning classifiers

- Lexicon-based approach
  - dictionary-based approach

- Corpus-based approach
  - statistical
  - semantic